

Package ‘RGS’

December 19, 2024

Title Recursive Gradient Scanning Algorithm

Version 1.0

Description Provides a recursive gradient scanning algorithm for discretizing continuous variables in Logistic and Cox regression models. This algorithm is especially effective in identifying optimal cut-points for variables with U-shaped relationships to 'lnOR' (the natural logarithm of the odds ratio) or 'lnHR' (the natural logarithm of the hazard ratio), thereby enhancing model fit, interpretability, and predictive power. By iteratively scanning and calculating gradient changes, the method accurately pinpoints critical cut-points within nonlinear relationships, transforming continuous variables into categorical ones. This approach improves risk classification and regression analysis performance, increasing interpretability and practical relevance in clinical and risk management settings.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports rms, SemiPar, survival

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Shuo Yang [aut, cre],
Yi Fei [aut],
Jinxin Zhang [ths]

Maintainer Shuo Yang <yangsh223@mail2.sysu.edu.cn>

Repository CRAN

Date/Publication 2024-12-19 20:50:02 UTC

Contents

evaluate_model_cox	2
--------------------	---

evaluate_model_logistic	2
find_best_combination_cox	3
find_best_combination_logistic	4
find_cutoffs_cox	4
find_cutoffs_logistic	5
rgs_for_cox	5
rgs_for_logistic	6
test_ushape_cox	7
test_ushape_logistic	8

Index 9

evaluate_model_cox	<i>Evaluate model with cutoffs Evaluates the model by creating exposure variables based on the cutoffs and fitting a Cox model.</i>
--------------------	---

Description

Evaluate model with cutoffs Evaluates the model by creating exposure variables based on the cutoffs and fitting a Cox model.

Usage

```
evaluate_model_cox(my_data, cutoffs, formula)
```

Arguments

my_data	A dataframe containing the data.
cutoffs	A list of dataframes containing the cutoffs for each variable.
formula	An object of class "formula" describing the model structure.

Value

A dataframe summarizing the model evaluation metrics.

evaluate_model_logistic	<i>Evaluate logistic regression model with cutoffs</i>
-------------------------	--

Description

This function evaluates the logistic regression model by creating exposure variables based on the cutoffs and fitting the model.

Usage

```
evaluate_model_logistic(my_data, cutoffs, formula)
```

Arguments

- my_data A dataframe containing the data.
- cutoffs A list of dataframes containing the cutoffs for each variable.
- formula An object of class "formula" describing the model structure.

Value

A dataframe summarizing the model evaluation metrics.

find_best_combination_cox

Find best combination of cutoffs Searches for the best combination of cutoffs that minimizes the Akaike Information Criterion (AIC).

Description

Find best combination of cutoffs Searches for the best combination of cutoffs that minimizes the Akaike Information Criterion (AIC).

Usage

```
find_best_combination_cox(my_data, cutoffs_list, formula)
```

Arguments

- my_data A dataframe containing the data.
- cutoffs_list A list of dataframes containing the cutoffs for each variable.
- formula An object of class "formula" describing the model structure.

Value

A dataframe containing the optimal cutoff values and the AIC value.

```
find_best_combination_logistic
```

Find best combination of cutoffs for logistic regression

Description

This function searches for the best combination of cutoffs that minimizes the Akaike Information Criterion (AIC).

Usage

```
find_best_combination_logistic(my_data, cutoffs_list, formula)
```

Arguments

my_data	A dataframe containing the data.
cutoffs_list	A list of dataframes containing the cutoffs for each variable.
formula	An object of class "formula" describing the model structure.

Value

A dataframe containing the optimal cutoff values and the AIC value.

```
find_cutoffs_cox
```

Find symmetric cutoffs This function finds symmetric cutoff points for a specified variable using Cox regression.

Description

Find symmetric cutoffs This function finds symmetric cutoff points for a specified variable using Cox regression.

Usage

```
find_cutoffs_cox(my_data, variable_name)
```

Arguments

my_data	A dataframe containing the data.
variable_name	A character string specifying the name of the variable for which to find cutoffs.

Value

A dataframe containing the symmetric cutoffs and their corresponding predicted log hazard ratios.

find_cutoffs_logistic *Find symmetric cutoffs for logistic regression*

Description

This function finds symmetric cutoff points for a specified predictor variable using logistic regression.

Usage

```
find_cutoffs_logistic(my_data, variable)
```

Arguments

my_data	A dataframe containing the data.
variable	A character string specifying the name of the predictor variable for which to find cutoffs.

Value

A dataframe containing the symmetric cutoffs and their corresponding predicted log odds ratios.

rgs_for_cox	<i>Main function for cox regression analysis Performs regression analysis to identify optimal cutoffs and evaluate the model. Users are required to create and set the datadist option globally before using this function.</i>
-------------	---

Description

Main function for cox regression analysis Performs regression analysis to identify optimal cutoffs and evaluate the model. Users are required to create and set the datadist option globally before using this function.

Usage

```
rgs_for_cox(my_data, u_variables, covariates)
```

Arguments

my_data	A dataframe containing the data.
u_variables	A character vector specifying the names of the U-shaped variables.
covariates	A character vector specifying the names of the covariates.

Value

A dataframe containing the best combination of cutoffs and associated AIC value.

Examples

```
library(rms)

data_path <- system.file("extdata", "cox_data.csv", package = "RGS")
my_data <- read.csv(data_path)

# Define variables
u_variables <- c("u1", "u2")
covariates <- c("l1", "c1")

# create and set the datadist option globally
dd <- datadist(my_data)
options(datadist = "dd")

rgs_for_cox(my_data, u_variables, covariates)
```

rgs_for_logistic *Main function for logistic regression analysis*

Description

This function performs logistic regression analysis to identify optimal cutoffs and evaluate the model. Users are required to create and set the datadist option globally before using this function.

Usage

```
rgs_for_logistic(my_data, u_variables, covariates)
```

Arguments

my_data	A dataframe containing the data.
u_variables	A character vector specifying the names of the U-shaped variables.
covariates	A character vector specifying the names of the covariates.

Value

A dataframe containing the best combination of cutoffs and associated AIC value.

Examples

```
library(rms)

data_path <- system.file("extdata", "logistic_data.csv", package = "RGS")
my_data <- read.csv(data_path)

# Define variables
```

```
u_variables <- c("u1", "u2")
covariates <- c("l1", "c1")

# create and set the datadist option globally
dd <- datadist(my_data)
options(datadist = "dd")

rgs_for_logistic(my_data, u_variables, covariates)
```

test_ushape_cox	<i>Test U-shape relationship This function checks for a U-shaped relationship between survival time and a set of variables. Users are required to create and set the datadist option globally before using this function.</i>
-----------------	---

Description

Test U-shape relationship This function checks for a U-shaped relationship between survival time and a set of variables. Users are required to create and set the datadist option globally before using this function.

Usage

```
test_ushape_cox(my_data, variables)
```

Arguments

my_data	A dataframe containing the data with survival times and event indicators.
variables	A character vector specifying the names of the variables to be tested.

Value

No return value, produces a plot as a side effect.

Examples

```
library(rms)

data_path <- system.file("extdata", "cox_data.csv", package = "RGS")
my_data <- read.csv(data_path)

# create and set the datadist option globally
dd <- datadist(my_data)
options(datadist = "dd")

test_ushape_cox(my_data, c("u1", "u2"))
```

test_ushape_logistic *Test U-shape relationship in logistic regression*

Description

This function plots the estimated relationship between the response variable and a set of predictor variables to identify potential U-shaped patterns. Users are required to create and set the `datadist` option globally before using this function.

Usage

```
test_ushape_logistic(my_data, variables)
```

Arguments

<code>my_data</code>	A dataframe containing the data with response and predictor variables.
<code>variables</code>	A character vector specifying the names of the predictor variables to be tested.

Value

No return value, produces a plot as a side effect.

Examples

```
library(rms)

data_path <- system.file("extdata", "logistic_data.csv", package = "RGS")
my_data <- read.csv(data_path)

# create and set the datadist option globally
dd <- datadist(my_data)
options(datadist = "dd")

test_ushape_logistic(my_data, c("u1", "u2"))
```


Index

`evaluate_model_cox`, [2](#)

`evaluate_model_logistic`, [2](#)

`find_best_combination_cox`, [3](#)

`find_best_combination_logistic`, [4](#)

`find_cutoffs_cox`, [4](#)

`find_cutoffs_logistic`, [5](#)

`rgs_for_cox`, [5](#)

`rgs_for_logistic`, [6](#)

`test_ushape_cox`, [7](#)

`test_ushape_logistic`, [8](#)